

Reporte Individual - Juego Flash Point Fire Rescue

Modelación de Sistemas Multiagentes con Gráficas Computacionales



Hecho por:

Rommel Toledo - A01709922

¿Por qué seleccionaron el modelo multiagentes utilizado?

Seleccionamos un modelo multiagentes de mesa con espacio de modelo MultiGrid y activación de agentes RandomActivation debido a las restricciones que las reglas originales del juego imponían, como lo es el hecho de que dos Bomberos (agentes en nuestro caso) pueden ubicarse en la misma casilla o que el juego se da por turnos, es decir, los agentes no pueden actuar a la vez en el juego (de aquí la utilidad de Random Activation).

El modelo que nosotros creamos permite modelar el entorno como existiría en el juego, esto debido a que al imponer las mismas restricciones y libertades, naturalmente se generan los escenarios que podrían ocurrir. Con la ventaja o desventaja de que los bomberos actuarían distinto a como actuarían de ser controlados por una persona.

Para que fuera fiel a las restricciones que tenemos usamos una combinación de variables y funciones para poder emular las reglas que se tienen.

¿Cuáles fueron las variables que se tomaron al momento de tomar la decisión?

A continuación se da una explicación de las variables que determinaron el diseño de nuestro modelo:

- **Distribución del terreno (Grid):** El espacio del tablero lo representamos mediante una cuadrícula, esta cuadrícula facilitó el gestionamiento de los elementos del juego al igual que de los bomberos y las mecánicas del juego. Al tener lugares definidos también facilitaba el poder encontrar objetos. Es la base del juego.
- **Puertas y Paredes:** Estos son uno de los principales limitantes de movimiento de los agentes, al toparse una pared deben romperla si quieren pasar rápidamente al otro lado, y al toparse una puerta deben abrirla primero para poder cruzarla. La distribución de estos puede beneficiar o perjudicar ciertas estrategias.
- **Puntos de Acción (AP):** Los bomberos tenían un límite fijo de cuantos puntos de acción tenían, con estos determinamos las acciones que pueden realizar en base a los puntos que poseen en el momento, las acciones son, moverse, apagar el fuego, rescatar víctimas, abrir o cerrar puertas y romper muros. La cantidad de estos determinan que tan sencillo o complicado puede serles ganar el juego.
- **Posiciones de Fuego y Humo:** Uno de los elementos principales en el juego se da por el peligro constante que es el humo, al iniciar el juego se pone fuego en ciertas casillas específicas y conforme pasa el tiempo se coloca humo en cada ronda, si cae al lado del fuego se prende, si cae sobre fuego explota y genera una reacción en cadena que destruye muros, puertas, daña bomberos y mata

víctimas. Esto juega un rol importante ya que es la causa de uno de los escenarios de derrota e influencia en cierta forma el comportamiento de nuestros agentes.

- Víctimas: Un POI puede ser una víctima potencial por lo que forman una parte importante del juego, los bomberos deben acercarse a determinar si se trata de una víctima o no y en caso de que sí buscar sacarla del mapa, es el elemento que permite el escenario de victoria, es decir, rescatar todas las víctimas.
- Condiciones del Juego: Las condiciones del juego determinan la victoria o derrota, si se salvan 7 víctimas se gana el juego, se pierden 4 víctimas también se pierde el juego, se juntan 24 puntos de daño se acaba el juego.
- Running: La utilidad de esta variable es registrar si el juego ha terminado o continua. Útil para poder detener la simulación.
- Cantidad de Bomberos: Al ser un juego cooperativo esto ayuda a determinar qué estrategias pueden ser más útiles que otras. Por ejemplo, el distribuir roles a cierto número de bomberos.

¿Cuál es la interacción de esas variables con respecto al resultado de la simulación?

Tal y como se mencionó anteriormente todas tienen su rol, y la combinación de los valores que pueden poseer es lo que determina qué tan sencillo o complicado es completar el juego exitosamente. Los principales determinantes son los que nombramos como “condiciones del juego”, esto a su vez dependen de otras variables y son las que determinan si se consigue la victoria o se cae en la derrota.

- La suma de daño acumulado se tiene ya sea cuando los bomberos rompen muros o el fuego lo hace, esto va acumulando gradualmente puntos hasta que puede llevar al edificio a derrumbarse.
- La cantidad de víctimas rescatadas es el determinante de victoria, este a su vez se ve afectado por las probabilidades de éxitos que tienen los bomberos de encontrar una víctima y sacarla del edificio, esta probabilidad a su vez varía dependiendo del fuego y puntos de acción de los bomberos principalmente.
- La cantidad de víctimas pérdidas, estas se pierden cuando el fuego llega a ellas directamente o llega a un bombero que las carga.

En general el grid sirve de base en donde se coloca todo, la distancia de las cosas respecto a los bomberos afectan cuantos AP deben gastar, los muros, puertas y fuego afectan por donde no pueden pasar, la posición de las víctimas y su cercanía a ellas determina igual la cantidad de AP a gastar. Es por eso que una buena estrategia es importante pues al tener

acciones limitadas es necesario que usen sus puntos de manera eficiente.

¿Por qué seleccionaron el diseño gráfico presentado?

El diseño gráfico fue estructurado siguiendo la arquitectura MVC y Clean Architecture. Su principal enfoque es simplificar la tarea de modelar los cambios del mapa, dado que recibimos el estado del mapa el cual incluye la ubicación de los muros, puertas, POIs, fuego, humo, y bomberos, lo que hacemos es instanciar los objetos una vez tenemos la información y ubicarlos en los espacios que corresponden a la ubicación que tienen en Python.

Este diseño gráfico fue seleccionado debido a que nos simplifica varias cosas, para efectos prácticos nos permitía tener la simulación del juego solo con cumplir las siguientes condiciones:

- Tener los elementos (piezas) que se ocupan en el juego. Esto fue sencillo ya que para los pisos y muros sólo los creamos y se les dió texturas importadas de tiendas de assets. Para las puertas y objetos del entorno como el fuego, humo y POIs, al igual que los bomberos usamos assets de Unity Store.
- Tener el código que permite generar el terreno.
- Tener el código que genera los muros y puertas.
- Tener el código que genera los elementos del entorno al igual que bomberos.

Teniendo esto era el equivalente de tener las piezas y únicamente tener que armar el mapa en base a las instrucciones que obtenemos de Python.

¿Cuáles son las ventajas que encuentras en la solución final presentada?

Las ventajas que encuentro son el hecho de que al llevar la lógica y reglas en Python, simplificamos enormemente la tarea de darle una representación visual, pues como explicamos anteriormente, a Unity únicamente le pasamos las instrucciones de cómo debía verse el mapa en ese momento y Unity se encargaba de crearlo. Es decir, al descomponer el problema en las dos principales secciones simplificamos la tarea de resolverlo, uno de nosotros se enfocó en la lógica del juego y bomberos, el otro en la representación visual. Esto permite que de ser necesario se puedan hacer pequeños ajustes en la estructura del juego desde Python y los cambios los veamos reflejados en Unity.

En cuanto a la estrategia de resolución hablamos, la principal ventaja que tiene (que también es su desventaja como hablaremos más adelante) es su simplicidad, pues fue fácil de implementar y es sencillo modificarla, se basa en que los bomberos pueden moverse de manera aleatoria con un sesgo dependiendo de la situación en que se encuentren, si poseen una víctima buscarán dejarla, si ven fuego apagarlo, y si encuentran un POI descubrirlo. Esto da pie a que, si bien es improbable, puedan generarse escenarios en que los bomberos se muevan de una manera en que parezcan responder a su entorno, como moverse en la dirección opuesta al fuego, acercarse a los POIs con víctimas, etc.

¿Cuáles son las desventajas que existen en la solución presentada?

Las desventajas que existen son el hecho de que nos acabamos especializando en la sección a la que nos enfrentamos al resolver el reto, esto hizo que casi no interviniéramos en lo que hacía el otro y por consecuencia se crearán errores que se pudieron fácilmente haber evitado si el otro estaba pendiente y daba su opinión al respecto. Debido a esto si bien simplificamos el resolver el problema, generamos muchos propios que nos robaron tiempo que pudimos dedicarle a pulir ya sea nuestra simulación o su visualización.

En cuanto a la estrategia para ganar el juego, está acabó siendo demasiado simplista impidiéndole a los agentes llegar a ganar, al ser aleatoria recibíamos resultados inconsistentes, en ocasiones perdíamos en 35 pasos en otros hasta 60 pasos, pero nunca llegábamos a rescatar a las 7 víctimas mínimas que ocupamos, esto se debe sobretodo a que los movimientos no son eficientes, y si bien están sesgados en base a ciertas estrategias como priorizar dejar a las víctimas, apagar fuego y descubrir POIs, no son lo suficientemente buenas como para permitir una victoria completa. Eventualmente o morían las víctimas o se derrumbaba el edificio.

¿Qué modificaciones podrías hacer para reducir o eliminar las desventajas mencionadas?

En cuanto a la estructuración del trabajo que afectó nuestros resultados yo hubiera hecho una planeación más flexible para no sobrecargarnos de trabajo y para haber planeado un cierto margen de tiempo que tuviéramos para cumplir las metas en lugar de una deadline fija que muchas veces no pudimos cumplir.

En cuanto a la estrategia yo buscaría utilizar una mejor estrategia que diera una mayor eficiencia, específicamente una basada en encontrar la distancia más corta para poder haber hecho priorizaciones por agentes, y hubiera tomado el hecho de que en el juego original se saben las posiciones en todo momento de los elementos ambientales como fuego, pois, humo, etc.

Al tener 6 agentes yo hubiera sentido más conveniente dividirlos en grupos y mediante un algoritmo como Dijkstra acercarlos a sus objetivos de manera eficiente.

Los grupos podrían haber sido:

Exploración: Se iban a enfocar en encontrar víctimas y una vez las hallaran buscarían rescatarlas evadiendo el fuego.

Eliminación de riesgos: Se enfocarían en apagar los fuegos que tuvieran mayor probabilidad de expandirse, esto lo detectaría en base a la distancia entre fuego al otro de manera horizontal y vertical.

Jack of all trades: Estos se encargarían de apoyar en las tareas más urgentes, si llegará a haber arriba de cierto nivel de llamas buscaría apagar llamas que tuvieran fuego de tres de sus lados, si no hubiera de

dos y así. Si un bombero cargando una víctima fuera en su dirección y ellos estuvieran más cerca de la puerta le quitarían la víctima y buscarían dejarla.